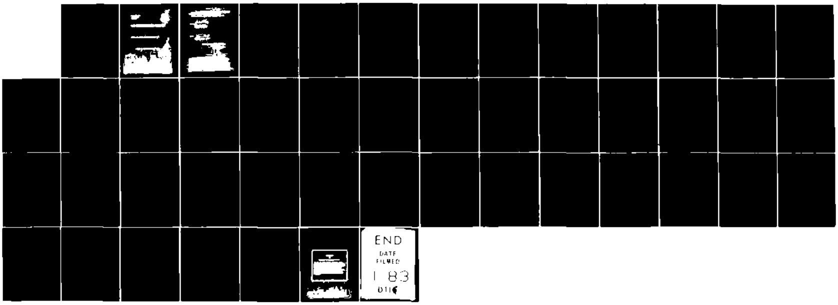
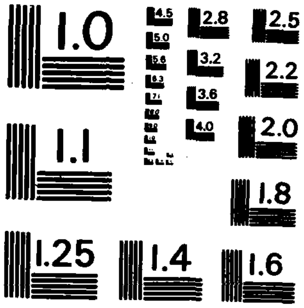
AD-A122 418 FAULT DETECTION/ISOLATION VERIFICATION(U) ROME AIR DEVELOPMENT CENTER GRIFFISS AFB NY W T ETTER ET AL. AUG 82 RADC-TR-82-232 1/1

UNCLASSIFIED F/G 12/1 NL




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER RADC-TR-82-232	2. GOVT ACCESSION NO. AD-A122418	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) FAULT DETECTION/ISOLATION VERIFICATION		5. TYPE OF REPORT & PERIOD COVERED In-House Report
		6. PERFORMING ORG. REPORT NUMBER N/A
7. AUTHOR(s) William T. Etter, RADC Robert A. Meyer, Clarkson College David E. Krzysiak, RADC		8. CONTRACT OR GRANT NUMBER(s) N/A
9. PERFORMING ORGANIZATION NAME AND ADDRESS Rome Air Development Center (DCLD) Griffiss AFB NY 13441		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 33126F 21550203
11. CONTROLLING OFFICE NAME AND ADDRESS Rome Air Development Center (DCLD) Griffiss AFB NY 13441		12. REPORT DATE August 1982
		13. NUMBER OF PAGES 52
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Same		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Fault Isolation
Performance Assessment
Digital Transmission
Communication Networks

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report documents an in-house program to test and evaluate the Fault Detection and Isolation Algorithm developed, by GTE Sylvania under RADC contract F30602-76-C-0433. Testing of the fault isolation capability required extensive use of the accompanying emulation facility, also developed by GTE. Several network connectivities were simulated. These included a Highly Interconnected Network, Dumbell Structured Network, Loop Network and a Backhauling Network. To test the performance of the (over)

Cont U

algorithm on these netowrks, several different fault scenarios were designed for each network. Each scenario would take advantage of the connectivity particular to each network. Single fault scenarios consisted of a single fault along with any logical sympathetic alarms that would be generated by such a fault. Multiple faults included several real faults along with a combination of sympathetic alarms that would be generated by the faults. The primary performance measures were accuracy of fault identification, memory usage of the algorithm and network files and fault identification time.

The results of the testing show that the algorithm is able to correctly identify the location of a fault to the equipment level under a wide range of conditions. It is noted, however, that the time required to find these faults varies considerably with network connectivity. In each network tested, the algorithm and network required less than 5550000 bytes of memory storage. Based upon the test results detailed in this report, it is recom ended that although the algorithm should be kept functionally the same, it should be rewritten to correct various deficiencies (such as execution time varying greatly with connectivity).

TABLE OF CONTENTS

1.0 SCOPE

This document describes the Fault Detection Isolation Verification (FDIV) testing conducted at the RADC System Control Laboratory from February 1980 to February 1981. The purpose of these tests is twofold, first to investigate the performance of the fault isolation algorithm developed by GTE as part of the CPMAS program, and secondly to determine what modifications and/or enhancements are necessary to provide an algorithm which may readily be integrated into the DCS.

The testing was conducted in accordance with the test plans and procedures contained in references 2b, using the network connectivities and station types described in reference 2a.

2.0 APPLICABLE DOCUMENTS

a. Fault Scenarios for Testing the CPMAS Fault Isolation Algorithm

b. Statement of Work - Evaluation of the GTE Fault Isolation Algorithm

c. Automated Performance Monitoring and Assessment for DCS Digital Systems - Final Technical Report July 1980 RADC-TR-80-196

d. CPMAS Emulator User's Manual - Nov 79

e. System Integration and Field Demonstration Test Plan/Procedures CDRL Sequence No. B019

f. Program Management and Implementation Plan: Digital European Backbone (DEB) Program, Electronic Systems Division Air Force Systems Command, Hanscom Air Force Base, Mass., 12 April 1976

g. Algorithms for Fault Detection and Isolation on Time Division Multiplexed Transmission Facilities, C.A. Danielson, MITRE Working Paper WP-21470, 9 December 1977

h. Digital Network Control, GTE Final Report for contract DCA100-76-C-0064, Defense Communications Engineering Center

3.0 OBJECTIVE

There were two primary objectives for this effort. The first was to investigate the performance of the fault isolation algorithm developed by GTE as part of the Communications Performance Monitoring and Assessment System (CPMAS) program. The second was to determine what modifications and/or enhancements would be necessary to provide an algorithm which could readily be integrated into the DCS.

The performance of the algorithm was measured by a set of tests performed at RADC using the CPMAS emulator. The primary performance measures evaluated were accuracy, execution time, and memory space requirements. Parameters which were varied to change the operating conditions included network size and topology, availability of monitor points, numbers and types of alarms, and operator input/output loading. The tests conducted in this program were designed to determine the sensitivity of the performance measures to variations in these parameters.

The second objective was met by an evaluation of the performance results and a comparison of these results with the requirements for operation within the DCS. A network was emulated which was drawn directly from the DCS environment (within the constraints of the emulator software). The purpose of that test phase was to demonstrate the applicability of the algorithm and associated performance data to the DCS, and to determine any modifications or enhancements to the fault isolation algorithm which should be made.

4.0 SYSTEM OVERVIEW

The Fault Detection/Isolation Verification (FDIV) test program was performed on the Communications Performance Monitoring and Assessment System (CPMAS) developed by GTE Sylvania under Contract No. F30602-76-C-0433. This section will describe the CPMAS equipment and explain its functional flow.

The CPMAS is a multiprocessor system. It consists of a PDP-11/60 minicomputer and two LSI-11/03 microprocessor systems. Most of the software on the 11/60 is written in FORTRAN, while the 11/03 software is written in MACRO-11, which is an assembly language. The PDP-11/60 processor would be located at larger manned communications node in the DCS, while the LSI-11/03's would be located at each station in the network.

The CPMAS system is logically broken down into three subsystems. The CPMAS emulation facility performs a network simulation function along with providing a low level data base generation capability. The Fault Detection and Isolation algorithm performs fault isolation on network information stored in the PDP-11/60, and can isolate faults to the equipment level. The CPMAS-D units are remote performance monitoring and assessment devices intended to be located at each station in the network. They collect fault information and provide communication between the Nodal Tech Control facility and each station.

4.1 The CPMAS Emulator

The CPMAS Emulator software resides in the PDP-11/60 computer system. The purpose of the Emulator is to provide a simulated Tech Control environment. The CPMAS Emulator performs the following key functions: Fault Detection and Isolation, Man/Machine (M/M) Interface, Station Emulation, Message Processing, and Data Base Generation.

4.1.1 The Data Base Generator

The Data Base Generator creates and manages all the necessary connectivity files required by the CPMAS Emulator. If a user wishes to represent a particulat network connectivity in the Emulator, he must input it via the Data Base Generator. First the user must draw out and detail the network on paper. It should be logically interconnected and completely specified. The user may then enter each station connectivity of the network. At the completion of each station entry, the Data Base generator will create the necessary station task image files. When all stations have been entered, the user must input a network file which specifies how each station is interconnected. At the completion of this phase, the Data Base Generator then creates all the necessary network task image files required by the Emulator. The data base generator allows the user to create networks consisting of up to 16 stations, 2 nodes and 22 links.

Figure 4-2 shows how data base access is controlled. The names ODBCNT, OEFS, OELFI, ONODLA, OSECTR and ONODLB are the actual names of the various Emulator software subroutines. ODBCNT is the data base controller. All I/O requests must be made through ODBCNT. OEFS and OELFI are subroutines from the Fault

Detection and Isolation algorithms. This figure shows how subroutines ONODLA, OSECTR, and ONODLB from the M/M interface require access to the data base via the data base controller.

4.1.2 Fault Detection and Isolation

The purpose of the Fault Detection and Isolation (FDI) algorithm is to delete sympathetic alarms from a communications network and correctly isolate real faults. The algorithm accomplishes this task by reading alarm status from an Equipment Alarm file. Then by going through a four step cycle, the algorithm will examine the status of the entire network, and locate all faulted equipment.

The first step of the FDI cycle is to translate the equipment alarm information into the corresponding network impact. This is accomplished by mapping each equipment alarm to hierarchy level (e.g., group 3), and by specifying the full hierarchical level (e.g., station ABC, link M0109, supergroup 2, group 3). This alarm information is maintained for use by FDI subroutines by the data base controller.

The second step of the FDI algorithm is to delete "sympathetic" alarms. This involves locating the furthest "upstream" alarmed communication hierarchy. Alarms in a communications network are propagated in a specific direction, either the receive (RX), or transmit (TX). If a TX fault is propagated in the RX direction, the upstream direction would be in the TX direction. The FDI algorithm assigns the furthest reported upstream alarm as the real alarm, and deletes the downstream alarms as sympathetic. The algorithm uses the network connectivity information that was previously defined by the user using the data base generator, and the alarm information generated by step one of the cycle to locate the furthest upstream station.

The third step of the algorithm is to identify the specific piece of faulted equipment in the communications hierarchy and to pass this information on to a display subroutine for operator viewing. The equipment identification is made through the use of the data base manager and the previously updated equipment alarm tables.

The final step of the FDI algorithm is to maintain the .100 most recently active faults in the modeled network. As fault reports are received, they are compared to existing reports. If they are new or depict a transition (e.g., from faulted to non-faulted), they are sorted for display. Any fault previously existing, but not currently reported is defined as intermittent and will be reported as such.

4.1.3 Man/Machine Interface (MMI)

The Man/Machine interface allows detailed status information about the network to be conveniently accessed for subsequent evaluation. This consists of Tech Controller commands, information displays, and information prompts. The commands allow tech controllers to acknowledge faults (ACK), clear unnecessary fault information from the screen (CLR), and to assign new thresholds for analog and pulse count alarm windows (ATH).

CPMAS EMULATION FACILITY

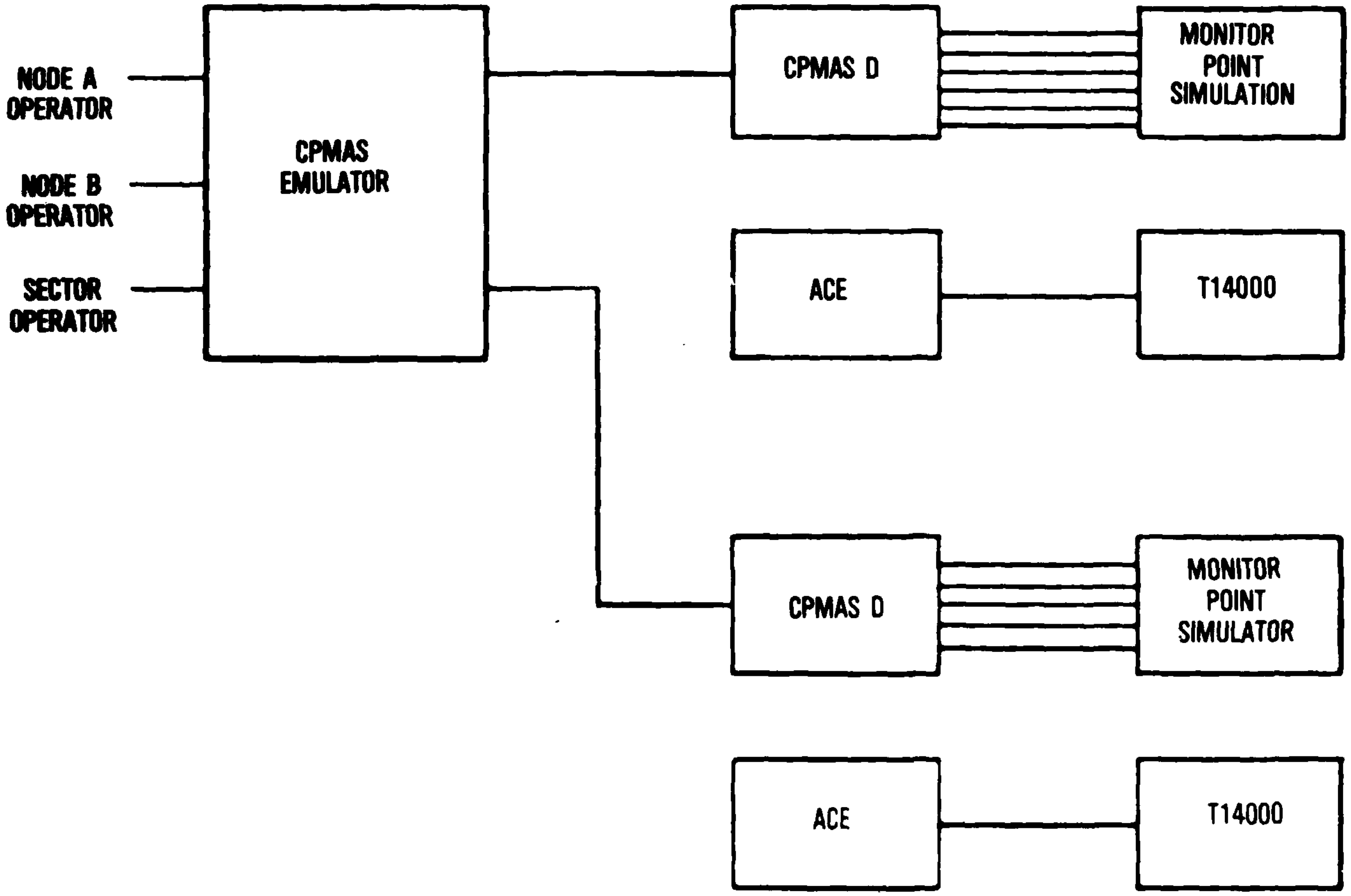


FIGURE 4-1 CPMAS EMULATION FACILITY

DATA BASE ACCESS CONTROL

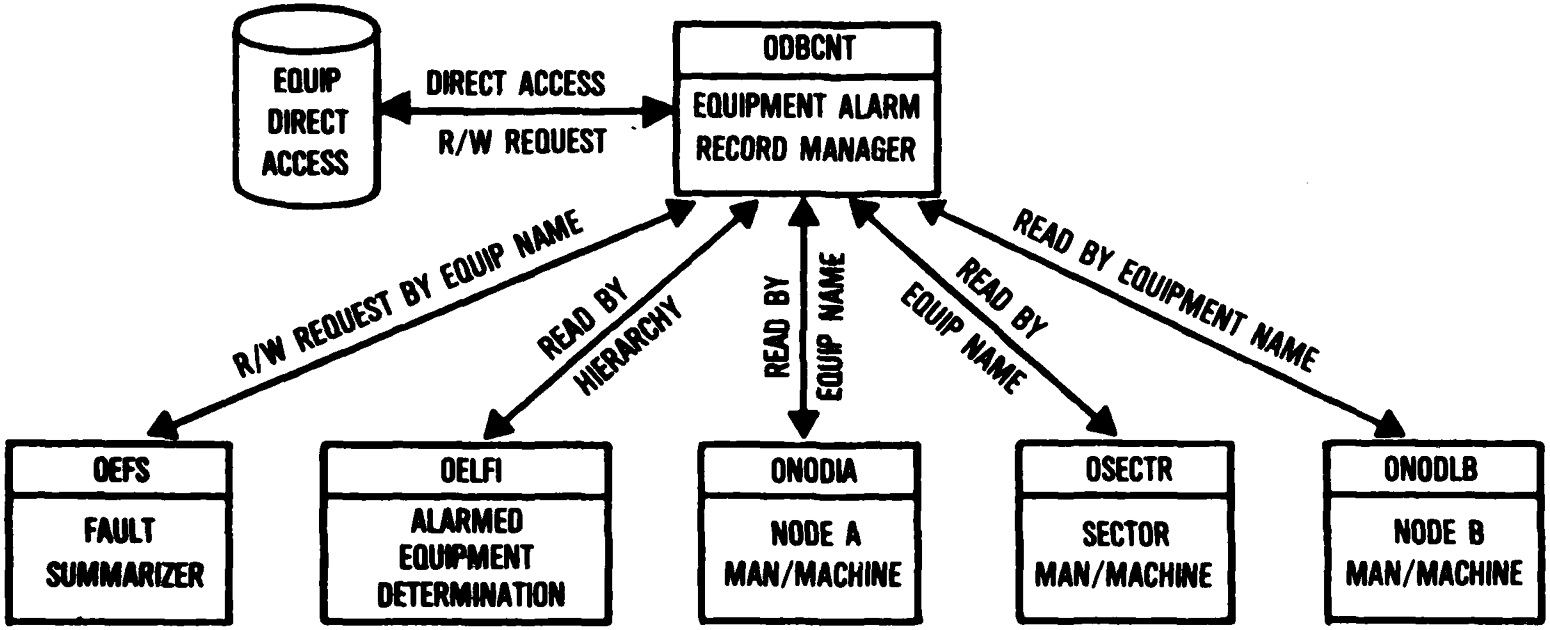


FIGURE 4-2 DATA BASE ACCESS CONTROL

There are four types of displays available to the tech controller: Fault Summary displays, Monitor Immediate displays, Equipment Detail displays and Threshold displays.

The Fault Summary display provides the tech controller with a listing of every faulty equipment in his area of responsibility. Through this display, the operator can determine the severity, location and status of the fault. The operator may then request an Equipment Detail display which will list all the alarmed monitor points in that station as reported to the CPMAS Emulator data base.

To find the current status of all monitor points in a particular station, the operator requests a Monitor Immediate display. This display lists all the monitor points of a particular piece of equipment in a station along with its status. If the monitor point is a analog or pulse count type then the current value of the monitor point at the time of the monitor immediate request is also displayed. To find out the current thresholds for a particular analog or pulse count monitor point, a Threshold display is used. It will display the red low, amber low, good, amber high and red high threshold values.

Information prompts aid the tech controller in the use of the system. When the fault displays have been udpated, the prompt FAULT will appear in the upper left hand corner of the VDU. When a message has been received from a CPMAS-D unit, a MESSAGE prompt will appear in the upper left hand corner. When a command is entered the operator receives a PENDING prompt on the command line until the Emulator takes an action. If the command is executed, then an ACK prompt is displayed. If the command is incorrectly specified, then ILL CMND is displayed on the command line.

4.1.4 Station Emulation

This function of the CPMAS Emulator uses the network and station connectivity files created by the user with the Data Base Generator. Using these files the Emulator models an actual communications network and allows the user to create faults in the network and use the fault detection isolation algorithm to delete sympathetic alarms and isolate faults. This is a very valuable tool in testing the fault detection algorithm.

The station emulation function works with the CPMAS-D units. The CPMAS-D units consist of a monitor point simulator, an Adaptive Channel Estimator (ACE) and a microcomputer which scans the monitor points and the ACE unit to check for alarmed conditions. The user can alarm the monitor point simulator or cause the ACE unit to be alarmed. These alarmed conditions are detected by the LSI 11/03 microcomputer and are reported via a communications port to the CPMAS emulator.

Faults may also be introduced into the model network by using the editor function that comes with DEC PDP 11/60 operating system (RSX-11M). Using the editor the user enters alarm information directly in a form that is compatible with the format of the CPMAS Emulator data base. Then executing a command while the fault isolation algorithm is operating causes the fault scenario to be executed. This is a powerful technique since faults may easily be introduced in any part of the network.

The CPMAS-D units represent only a subset of the total equipment in a station, and hence only alarms can be created in that particular subset when entering alarms through the CPMAS-D.

4.1.5 Message Processing

Message processing consists of handling all communications between the CPMAS Emulator and the CPMAS-D units. These communications can originate from two sources. First the CPMAS-D unit when it scans keeps a record of the current status of all monitor points. Whenever it detects a change in these monitor point (e.g., from alarmed to unalarmed or from unalarmed to alarmed) it formats and sends this information to the CPMAS emulator. Then this information is used to update the equipment status table in the emulator data base. The second type of communications is when the operator requests information about, or wants to change information in the CPMAS-D data base. The operator requests information about the data base by the Monitor Immediate (MIM) command. He can also display the current alarm threshold. The operator can change the data base by changing the values of the current alarm thresholds. Message processing then handles all communications to and from the CPMAS-D units.

5.0 DESCRIPTION OF TESTS

This section describes the tests that were performed by RADC in the Fault Detection/Isolation Verification in-house effort. The test program consisted of three major parts. For the first part, the tests were a subset of those performed by GTE as part of their system integration test. The purpose of repeating these tests was to verify correct system operation, and gain confidence with the use of the system. The test engineer selected a subset of the system integration tests and duplicated them in order to verify that these results agreed with the results obtained by GTE.

The second phase consisted of emulating four different model networks with a variety of fault scenarios. The performance data that was collected consisted of execution time and memory space requirements for each of the emulated fault scenarios. The fault display was compared with the expected results to verify correct isolation of the faults. In the case of discrepancy, a hard copy of the fault display was analyzed to determine the cause of the problem.

The third phase was an emulation of a model network which represents an actual nodal area as closely as possible. Since the fault isolation algorithm is currently implemented as a prototype version for test and demonstration purposes, there were constraints which limited the size and complexity of the network. However, the network was sufficiently representative of an actual nodal control area to demonstrate the capabilities and limitations of the fault isolation algorithm within the DCS environment.

5.1 FDIV Network Configurations

In order to demonstrate that the fault isolation algorithm is capable of operating properly over a wide range of network configurations, four model networks were carefully designed to determine the sensitivity of the algorithm to network structures of interest. The principal performance measure used in these tests was accuracy of the algorithm, i.e., does the algorithm correctly identify the fault condition and delete the sympathetic alarms. Additional performance measures included execution time and memory space utilization.

5.1.1 Highly Interconnected Network

A highly interconnected network is one in which the ratio of communication paths to stations is large. The network used during the testing is shown in Figure 5-1. It represents the worst case example of such a structure of similar size. By testing the performance of the fault isolation algorithm for this configuration and comparing the results with the results of testing other network configurations, the sensitivity of the algorithm to network connectivity was determined.

For this algorithm, this network model was an especially important test. The execution time of the algorithm is proportional to the square of the number of stations which are defined as subnode centers. In this context, a subnode center is any station with more than two links connected to it. Fpr this network every station was a subnode center, and thus this network was a worst case configuration of this size for the algorithm.

5.1.2 Dumbbell Structured Network

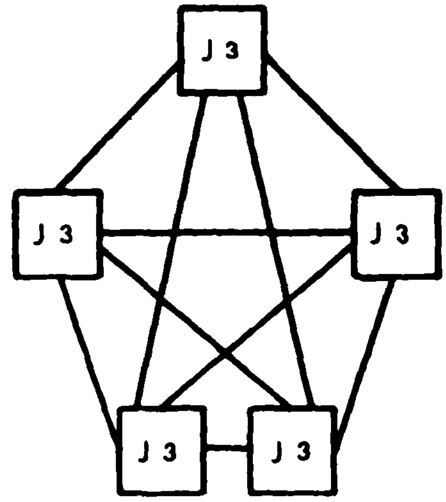
A dumbbell structured network is representative of a common configuration in the DEB network. As shown in Figure 5-2, it consists of a chain of repeater stations, called R, between two junction points. The size of the network may be varied by either inserting additional repeater stations, or by replacing any of the terminal stations by another junction with additional terminal stations. This structure provided an excellent environment for testing the ability of the fault isolation algorithm to properly detect the propagation of sympathetic alarms through several intervening, unalarmed stations. This network was used in third phase of testing due to its DEB configuration simularity.

5.1.3 Loop Network

The network shown in Figure 5-3 contains a group of stations which are interconnected by more than one communications path, thus forming a loop. This structure is representative of configurations found in DEB and could have presented difficulties for the fault isolation algorithm. The primary concern in testing this network was to determine the accuracy of the algorithm.

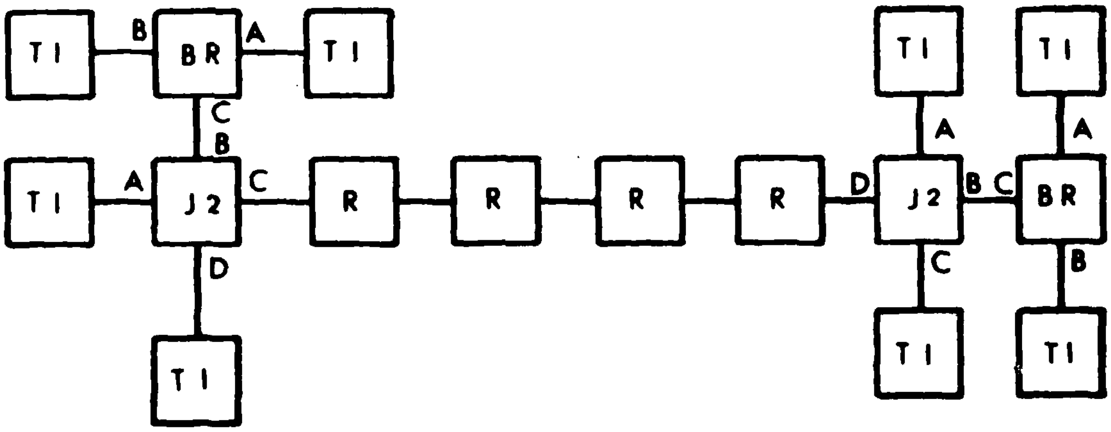
5.1.4 Backhauling Network

Backhauling is a technique to provide a channel between two points between which there is no direct available channel. Instead of a direct channel, a channel is made available via an indirect route. The network shown in Figure 5-4 is an example of backhauling through the terminal station T2. The distinction that we made in this test program between looping and backhauling is that the signal in a loop enters a station from one link and leaves from another link while in backhauling, the signal enters and leaves the station from the same link.



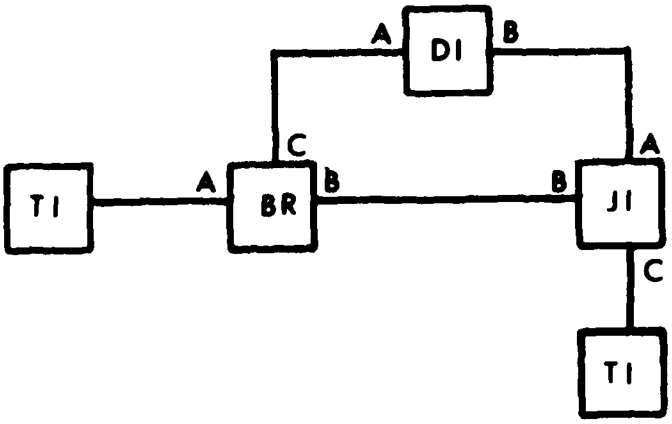
HIGHLY INTERCONNECTED NETWORK

FIGURE 5-1



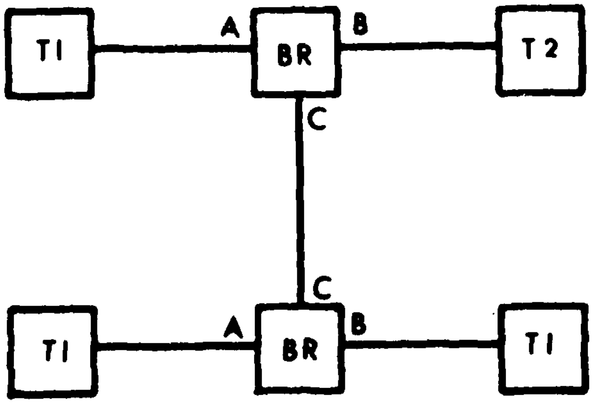
DUMBBELL STRUCTURED NETWORK

FIGURE 5-2



LOOP NETWORK

FIGURE 5-3

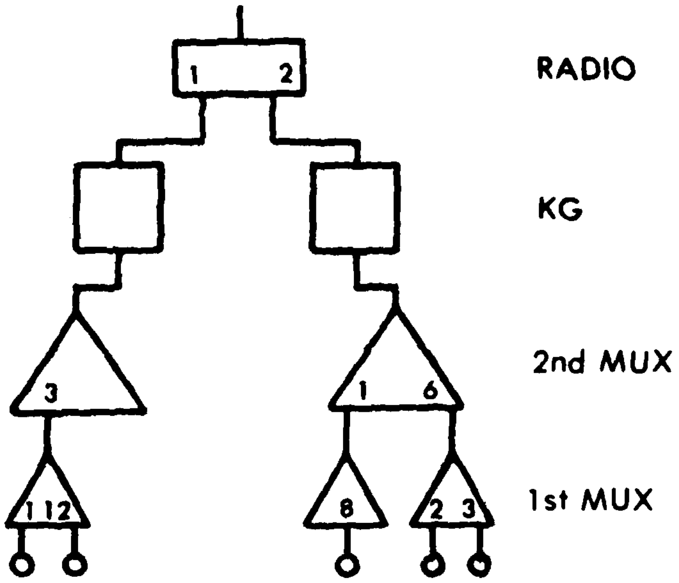


BACKHAULING NETWORK

FIGURE 5-4

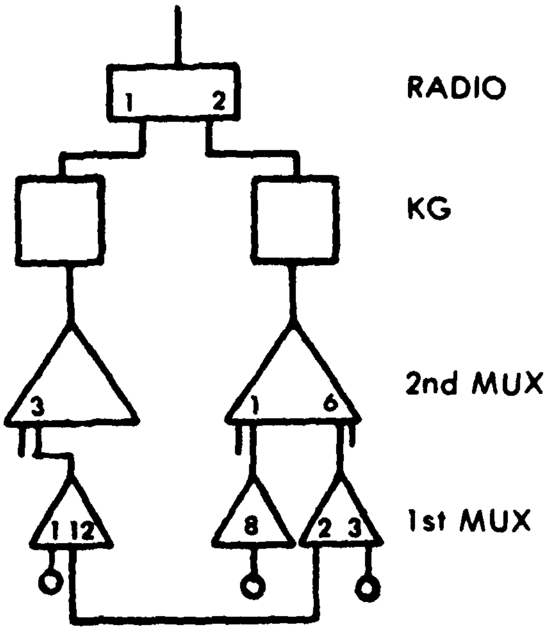
5.2 Station Configurations

As shown in the previous diagrams for each of the four model networks, a common set of station types have been used. These station types were selected to keep the emulation and network generation as simple as possible while providing the necessary variety in the network structures. The equipment configuration for each station type is shown in figures 5-5 through 5-12. In all cases the equipments were assumed to be DRAMA type equipments. Although the basic algorithm is not equipment-type dependent, the software as implemented for emulation purposes had no provisions for non-DRAMA equipments.



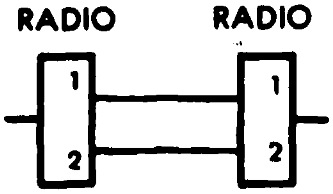
TI-TERMINAL STATION

FIGURE 5-5



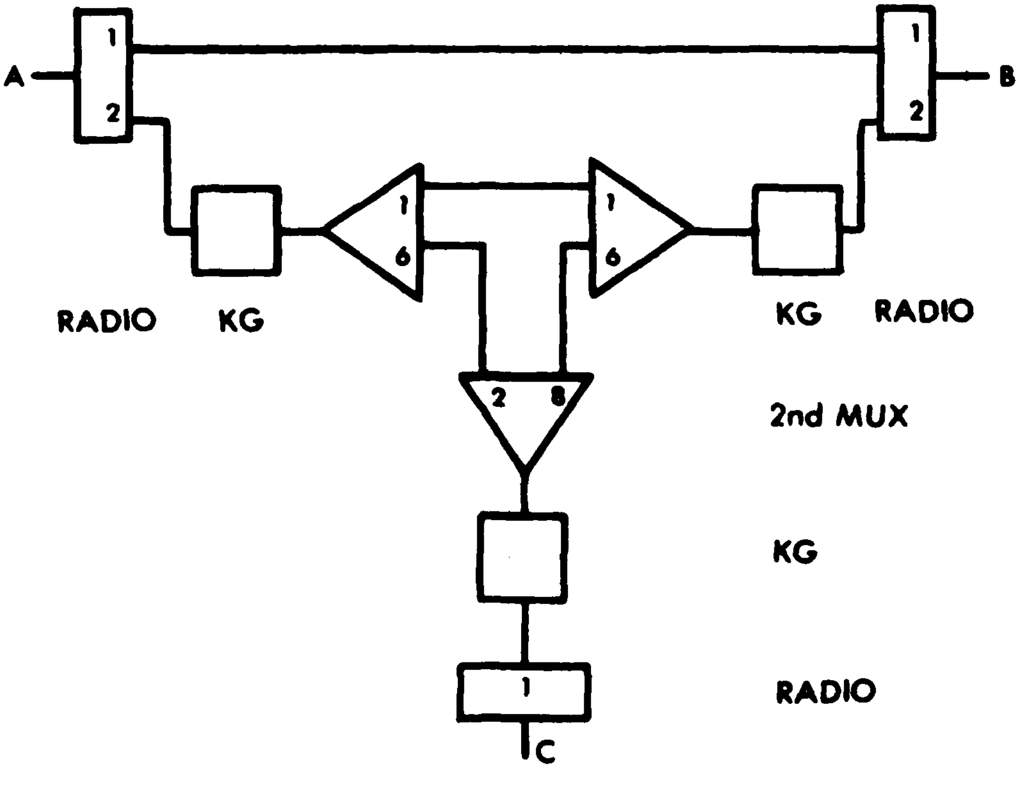
T2-TERMINAL STATION

FIGURE 5-6



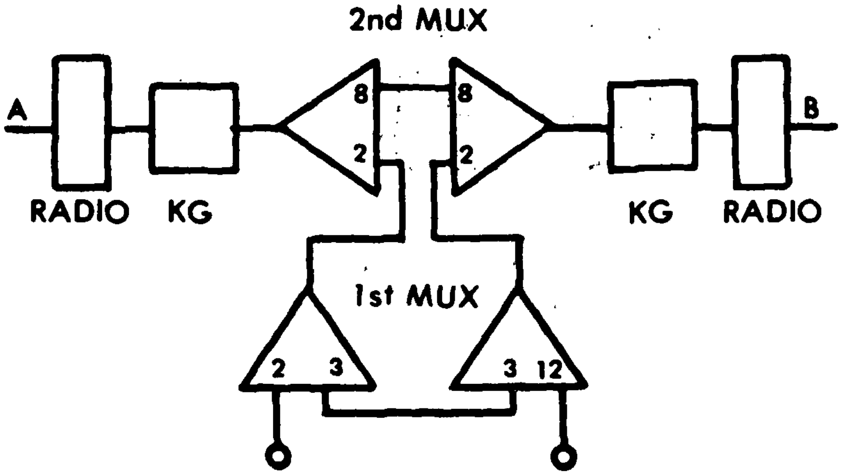
R-REPEATER STATION

FIGURE 5-7



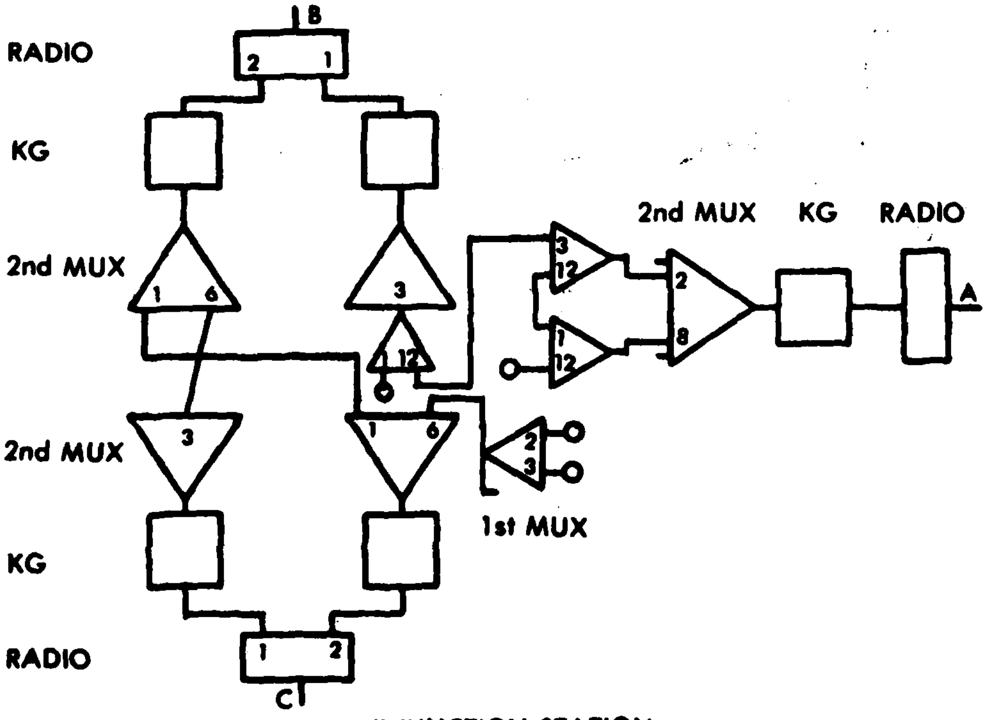
BR-BRANCHING REPEATER

FIGURE 5-8



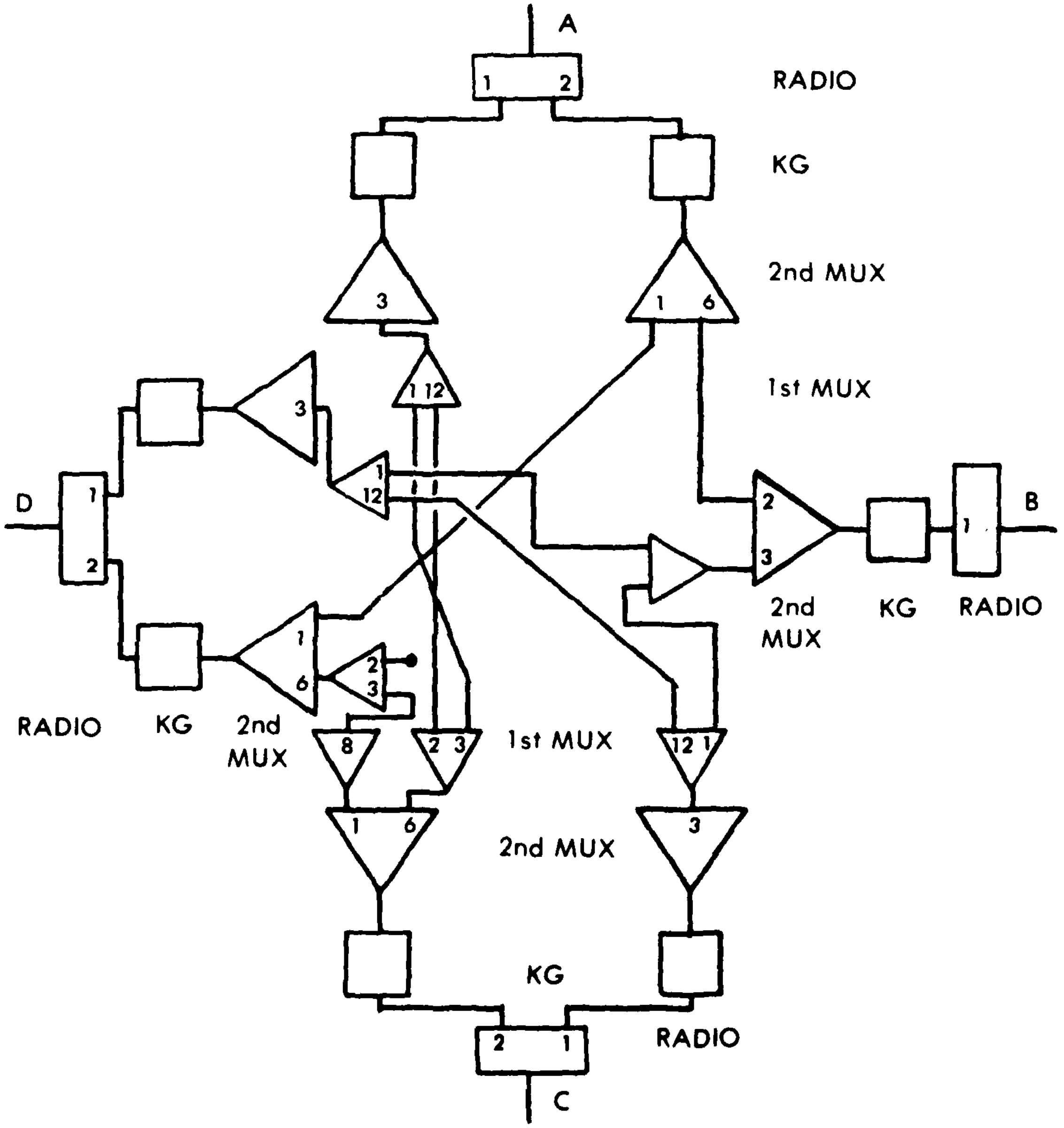
DI-DROP & INSERT STATION

FIGURE 5-9



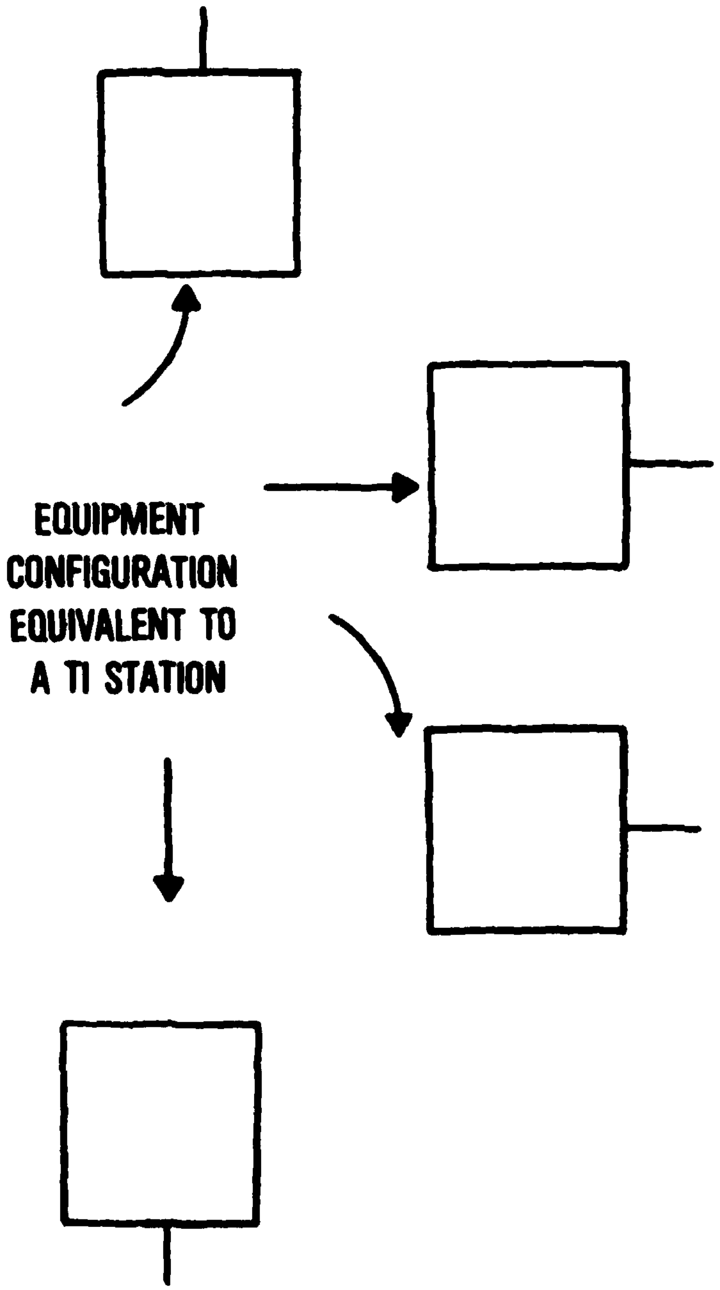
JI-JUNCTION STATION

FIGURE 5-10



JUNCTION STATION

FIGURE 5-11



J3 JUNCTION STATION

FIGURE 5-12

5.3 Fault Scenarios

This section describes the fault scenarios used in the testing of the Fault Isolation Algorithm. Each fault scenario consists of one or more faults. A fault is defined by the presence of an alarm for the faulty equipment and possibly the presence of additional alarms which are sympathetic with the faulty equipment.

In Section 5.3.2 we catalog the types of alarms used in the fault scenarios. In general, local alarms have been omitted since, by definition, such alarms are self-isolating. Alarms at each of the levels of the communications hierarchy (channel, group and supergroup) are included. Also included are alarms to be set at the CPMAS-D units on an intermittent basis. In Sections 5.3.3, 5.3.4, 5.3.5 and 5.3.6, faults are defined for each of the four networks tested. Each fault definition specifies the faulty equipment and associated alarm together with sympathetic alarms. Following this is a description of how to construct a variety of single fault scenarios using the fault definitions. Section 5.3.7 defines the multiple fault scenarios tested. Section 5.3.8 describes the fault scenarios for intermittent faults using the CPMAS-D units.

5.3.1 Preliminary Scenarios

Before the final set of scenarios could be constructed for testing the fault isolation algorithm on the four test networks, it was first necessary to gain some insight on exactly how sympathetic alarms were treated along a transmission path. We were also interested in how local alarms were differentiated from communication alarms and whether their severity changed with changing alarm conditions. In order to accomplish this, a set of scenarios were run and the results studied.

The Highly Interconnected Network was chosen for this testing since although it represents worst case as far as execution time goes, essentially it is still a very simple network, consisting only of four stations each composed of the equipment equivalent of four terminal stations. Stations QRS and NOP were chosen as a communications path, and various scenarios were executed around the following list of faults:

NOP		
F003	#18	Loss of CHTX Data or Timing - 2
	#29	Loss of GPTX Data
	#30	Loss of GPTX Timing
S002	#30	Loss of GPTX Data or Timing - A/B - 1
	#33	Loss of SGTX Data - A
	#35	Loss of SGTX Timing - A
R001	#20	Frequency Drift Alarm - A
	#22	Loss of Modulator Output - A
	#24	Loss of Multiplexer Output - A
	#27	Loss of MBS Port 2
	#30	Loss of Timing Port 2
	#42	Tx Power Level A Red

QRS

R004	# 1	Loss of Decoder Output - A
	# 3	Loss of Derandomizer Output - A
	# 5	Loss of Frame Synchronization - A
	# 9	Loss of MBS Port 2 - A
	#15	Loss of Timing Port 2 - A
	#35	Demux Frame Error A
	#38	Rx Signal Level A Red
	#54	SDR A Red
	#58	SNR A Red
	#62	BER A Red
S008	# 1	Loss of SGRX Frame - A
	# 3	Loss of SGRX Data - A
	# 5	Loss of SGRX Timing - A
	#12	Loss of GPRX Data or Timing - A - 6
	#23	Loss of GPRX Timing - A
	#42	SGRX Frame Error Alarm A
	#46	SGRX Frame Loss Alarm A
F012	# 1	Loss of GPRX Frame
	# 2	Loss of GPRX Data
	# 3	L oss of GPRX Timing
	# 5	Loss of CHRX Data or Timing -2
	#33	GPRX Frame Error Alarm
	#35	GPRX Frame Loss

The testing methodology for this set of scenarios consists of running the first scenario with all faults in NOP and QRS. Then gradually, all faults in NOP will be eliminated in future scenarios, until finally only QRS has alarms in it. Next 3 more scenarios will be run with all alarms in QRS, and a scenario with only one alarm in NOP, testing each of NOP's 3 station equipments.

5.3.2 Alarm Types

The table shown below lists all of the alarm types used in the following fault scenarios. The table is organized by channls, groups, and supergroups. For each alarm the direction (i.e., transmit or receive), the equipment, the name and the number are given. Equipment types are specified as: F-first multiplexer, S - second multiplexer, R - radio, K - KG-81. In some cases the alarm number depends on the channel, group or supergroup number; for these cases the alarm number is specified as: K+c or K+g or K+s, where c, g, and s represent the channel, group, or supergroup number, and K is an integer constant.

	DIR	EQU	NAME	NUMBER
CHANNEL				
	TX	F	Loss of data or timing	16+c
	RX	F	Loss of data or timing	3+c

GROUP			
TX	F	Loss of data	29
TX	S	Loss of data or timing	24+g
RX	F	Loss of data	2
RX	S	Loss of data or timing	6+g
SUPERGROUP			
TX	S	Loss of data	33
TX	R	Loss of MBS	25+s
RX	S	Loss of data	3
TX,RX	K	Summary alarm	1
INTERMITTENT CHANNEL at QRS			
TX	F001	Loss of data or timing (binary)	17
INTERMITTENT GROUP at QRS			
RX	F001	Frame error (pulse count)	
INTERMITTENT SUPERGROUP at QRS			
TX	R001	Transmit power - A Amber (analog)	41

5.3.3 Faults in the Highly Interconnected Network

The following single faults are defined for the highly interconnected network. The format of the fault definition is: fault identifier (single letter, A-Z), station name, equipment name, and fault type. Following the fault definition is the list of alarms

A. Station HI1, first mux F001, TX channel 12 input.

Channel 12 TX on F001 at HI1	fault alarm
Channel 12 RX on F010 at HI3	symp. alarm

B. Station HI2, first mux F003, TX group output.

Group TX on F003 at HI2	fault alarm
Group 6 TX on S002 at HI2	symp. alarm
Group 6 RX on S008 at HI1	symp. alarm
Group RX on F012 at HI1	symp. alarm
Channel 2 RX on F012 at HI1	symp. alarm
Channel 3 RX on F012 at HI1	symp. alarm

C. Station QRS, second mux S008, TX supergroup output.

Supergroup TX on S008 at QRS	fault alarm
Supergroup TX,RX on KG8 at QRS	symp. alarm
Supergroup 2 TX on R004 at QRS	symp. alarm
Supergroup RX on S002 at NOP	symp. alarm
Group 1 RX on S002 at NOP	symp. alarm
Group 6 RX on S002 at NOP	symp. alarm
Group RX on F002 at NOP	symp. alarm
Group RX on F003 at NOP	symp. alarm
Channel 8 RX on F002 at NOP	symp. alarm
Channel 2 RX on F003 at NOP	symp. alarm
Channel 3 RX on F003 at NOP	symp. alarm

D. Station NOP, KG2, summary alarm

Supergroup TX,RX on KG2 at NOP	fault alarm
Supergroup RX on S002 at NOP	symp. alarm
Group 1 RX on S002 at NOP	symp. alarm
Group 6 RX on S002 at NOP	symp. alarm
Group RX on F002 at NOP	symp. alarm
Group RX on F003 at NOP	symp. alarm
Channel 8 RX on F002 at NOP	symp. alarm
Channel 2 RX on F003 at NOP	symp. alarm
Channel 3 RX on F003 at NOP	symp. alarm
Supergroup RX on S008 at QRS	symp. alarm
Group 1 RX on S008 at QRS	symp. alarm
Group 6 RX on S008 at QRS	symp. alarm
Group RX on F011 at QRS	symp. alarm
Group RX on F012 at QRS	symp. alarm
Channel 8 RX on F011 at QRS	symp. alarm
Channel 2 RX on F012 at QRS	symp. alarm
Channel 3 RX on F012 at QRS	symp. alarm

The primary feature of the fault isolation algorithm investigated using the highly interconnected network was execution time, since this network represents a worst case configuration. For single fault tests, four fault scenarios were prepared corresponding to faults a,b,c and d above. These four scenarios cover each of the levels of communications hierarchy and also cover a wide range of the number of alarms to be processed. In addition, other fault scenarios were run in which not all of the sympathetic alarms were present. The purpose of such scenarios was to verify correct isolation in the absence of some of the expected alarms.

As a test of the response when the actual fault alarm is missing and only the sympathetic alarms were present, a fault scenario was prepared using fault C with the three alarms at QRS omitted. This simulates a situation in which alarm data is not available from station QRS. A similar test was made using fault D with only the fault alarm (on the KG) omitted and all sympathetic alarms present. This is an interesting case since the sympathetic alarms are available on both sides of the fault equipment even when the actual fault alarm is not present.

5.3.4 Faults in the Dumbbell Structured Network

The following single faults are defined using the same format as in the previous section.

A. Station J2A, second mux S005, TX group 6 input.

Group 6 TX on S005 at J2A	fault alarm
Group 6 RX on S004 at NOP	symp. alarm
Group RX on F003 at NOP	symp. alarm
Channel 2 RX on F003 at NOP	symp. alarm
Channel 3 RX on F003 at NOP	symp. alarm
Channel 8 TX on F004 at NOP	symp. alarm
Channel 8 RX on F002 at QRS	symp. alarm

B. Station T1E, first mux F003, TX channel 3 input.

Channel 3 TX on F003 at T1E	fault alarm
Channel 3 RX on F007 at NOP	symp. alarm
Channel 1 TX on F006 at NOP	symp. alarm
Channel 1 RX on F001 at QRS	symp. alarm

C. Station BRA, second mux S002, RX supergroup input.

Supergroup RX on S002 at BRA	fault alarm
Group 1 RX on S002 at BRA	symp. alarm
Group 6 RX on S002 at BRA	symp. alarm
Group 1 TX on S001 at BRA	symp. alarm
Group 8 TX on S003 at BRA	symp. alarm
Group 1 RX on S002 at T1B	symp. alarm
Group RX on F002 at T1B	symp. alarm
Channel 8 RX on F002 at T1B	symp. alarm
Group 8 RX on S007 at J2A	symp. alarm
Group RX on F007 at J2A	symp. alarm
Channel 2 RX on F007 at J2A	symp. alarm
Channel 3 RX on F007 at J2A	symp. alarm
Channel 1 TX on F006 at J2A	symp. alarm
Channel 1 TX on F002 at J2A	symp. alarm
Channel 1 RX on F002 at NOP	symp. alarm
Channel 2 TX on F007 at NOP	symp. alarm
Channel 2 RX on F003 at T1E	symp. alarm
Channel 1 RX on F001 at T1D	symp. alarm

D. Station T1C, first mux F001, TX channel 1 input.

Channel 1 TX on F001 at T1C	fault alarm
Channel 1 RX on F001 at J2A	symp. alarm
Channel 3 TX on F005 at J2A	symp. alarm
Channel 3 RX on F003 at NOP	symp. alarm
Channel 8 TX on F004 at NOP	symp. alarm
Channel 8 RX on F002 at QRS	symp. alarm

E. Station NOP, first mux F004, RX channel 8 output.

Channel 8 RX on F004 at NOP	real alarm
Channel 3 TX on F003 at NOP	symp. alarm
Channel 3 RX on F005 at J2A	symp. alarm
Channel 1 TX on F001 at J2A	symp. alarm
Channel 1 RX on F001. at T1C	symp. alarm

The fault scenario using fault A is designed to test the ability of the fault isolation algorithm to recognize sympathetic alarms which are several stations downstream from the actual fault, especially when separated by stations having no alarms (the repeaters in this case). Variations of this fault scenario were run including cases: a) with some of the sympathetic alarms omitted; b) with the sympathetic alarms appearing prior in time to the fault alarm; and c) with all alarms at an intervening station (i.e., NOP) omitted.

The fault scenario consisting of fault B is designed to test the response of the algorithm when sympathetic alarms are present in a nodal area adjacent to the nodal area in which the fault occurs.

Fault C is a complex scenario which tests the ability of the algorithm to properly identify sympathetic alarms which propagate downstream from the fault in two or more different paths as a result of branch points in the network. Note that this scenario also includes alarms in both nodal areas.

The remaining two faults, D and E, are primarily intended for later use in multiple fault testing.

5.3.5 Faults in the Loop Network

The faults for the loop network are defined below following the format of the previous sections.

A. Station NOP, second mux S001, TX group 8 input.

Group 8 TX on S001 at NOP	fault alarm
Group 8 RX on S002 at LD1	symp. alarm
Group 8 TX on S001 at LD1	symp. alarm
Group 8 RX on S003 at LR1	symp. alarm
Group 6 TX on S002 at LR1	symp. alarm
Group 6 RX on S003 at NOP	symp. alarm
Group 3 TX on S004 at NOP	symp. alarm
Group 3 RX on S001 at QRS	symp. alarm
Group RX on F001 at QRS	symp. alarm
Channel 1 RX on F001 at QRS	symp. alarm
Channel 12 RX on F001 at QRS	symp. alarm

B. Station LT1, first mux F001, TX channel 12 input.

Channel 12 TX on F001 at LT1	fault alarm
Channel 12 RX on F002 at NOP	symp. alarm
Channel 3 TX on F001 at NOP	symp. alarm
Channel 3 RX on F002 at LD1	symp. alarm

Channel 3 TX on F001 at LD1	symp. alarm
Channel 3 RX on F003 at LT1	symp. alarm

Faults A and B each represent a fault scenario in which the out-of-service communication path forms a loop, that is, it returns to the originating station. The primary difference between these two cases is the level of the communication hierarchy assumed to be effected by the faulty equipment. Fault A is the group level, while fault B is at the channel level. The purpose of these scenarios is to determine if the algorithm will become "confused" or lost in a never ending execution loop as a result of such an unusual topology.

5.3.6 Faults in the Backhauling Network

In order to test the performance of the algorithm in the presence of backhauling, the following scenario was run on the backhauling network. Variations of this basic scenario were run, for example, with all alarms from an intervening station (i.e., QRS) omitted.

A. Station BT3, first mux, TX channel 2 input.

Channel 2 TX on F003 at BT3	fault alarm
Channel 2 RX on F003 at QRS	symp. alarm
Channel 12 TX on F001 at QRS	symp. alarm
Channel 12 RX on F001 at BT1	symp. alarm

5.3.7 Multiple Faults

There were four basic tests made using multiple fault scenarios. All of the multiple fault tests were run using the dumbbell structured network since it is the most general type of network and the largest network being simulated. In the first, the two faults are along separate paths and have no alarms or equipment in common. The second scenario tests for the ability of the algorithm to isolate two faults which have some sympathetic alarms in common, but which are not masked by one another. The third scenario tests for the ability to isolate two faults which share the same path (i.e., along the same link, supergroup, group, etc.), but in different directions. The last scenario involves two faults in which one of the faults is masked (i.e., appears to be a sympathetic alarm) by the other. Only after the higher level fault is corrected, can the lower level fault be isolated.

A. Station T1A, second mux S002, TX group 1 input.

Group 1 TX on S002 at T1A	fault alarm
Group 1 RX on S002 at BRA	symp. alarm
Group 1 TX on S001 at BRA	symp. alarm
Group 1 RX on S002 at T1B	symp. alarm
Group RX on F002 at T1B	symp. alarm
Channel 8 RX on F002 at T1B	symp. alarm

Station J2A, second mux S005, TX group 6 input fault alarm
(See Scenario A under section 5.3.3 for alarm list)

B. Combination of 5.3.3-A and 5.3.3-D.

C. Combination of 5.3.3-D and 5.3.3-E.

D. Sequence consisting of 5.3.3-A followed by a scenario with the alarms shown below CLEARED.

Group 6 TX on S005 at J2A	fault alarm
Group 6 RX on S004 at NOP	symp. alarm

The remaining alarms are then:

Group RX on F003 at NOP	fault alarm
Channel 2 RX on F003 at NOP	symp. alarm
Channel 3 RX on F003 at NOP	symp. alarm
Channel 8 TX on F004 at NOP	symp. alarm
Channel 8 RX on F002 at QRS	symp. alarm

5.3.8 Intermittents fault and CPMAS-D Tests

These tests have two objectives. The first was to verify the capability of the fault isolation algorithm to identify and display intermittent faults. In conjunction with this test we wished to determine certain basic performance characteristics of this capability. Specifically, what is the minimum time duration in which the equipment must be alarmed in order to be identified; what is the maximum rate of occurrence which can be counted?

The second objective is to determine what effect, if any, increased message handling between nodal control and the CPMAS-D units has on the execution time of the fault isolation algorithm. Additional message handling was imposed on the system by requesting monitor immediate actions at the CPMAS-D units from nodal control. Such action is probably typical of the action taken by an operator during a time of crisis.

5.3.8.1 Intermittents

Since the objective of the intermittent tests was concerned with timing, these tests were performed on a single network rather than all four different networks. We did not expect a great deal of difference in performance as the network topology is varied. Since the highly interconnected network represents the expected "worst case" in terms of algorithm execution time, this network was used for the testing of intermittents.

Three intermittent alarms were defined in section 5.3.1 which include three alarm types - binary, pulse, and analog. For each alarm the test procedure was to vary the frequency and duration of alarm on/off cycles and to observe the system response. These tests were also run in conjunction with the previously defined fault scenarios (5.3.2A - 5.3.2D) for this network. Any differences, with respect to the previous executions of scenarios 5.3.2A - 5.3.2B, in system performance were noted.

5.3.8.2 CPMAS-D Tests

For each of the scenarios defined in sections 5.3.2 - 5.3.5, the impact of additional CPMAS-D message handling was investigated. This was done as follows: first, the scenario was run with no requests for data from the operator; the scenario was then repeated two or three times with an increasing number of monitor immediate requests by the operator. The parameter noted in these trails was the variation in fault isolation time as a function of the number of messages handled.

6.0 Test Results

This section contains the results of the testing of the Fault Detection and Isolation algorithm. Three types of data were collected: timing, accuracy and memory requirements. Timing is the length of time for one fault isolation cycle. Accuracy is whether the correct fault or faults were identified, and whether all the correct sympathetic alarms were deleted. Memory requirements are how much computer memory (in bytes) the program took up and how much memory each network connectivity required.

6.1 Algorithm Accuracy

During the testing of the algorithm, there were only three occasions in which the real fault was not correctly identified, and/or all the sympathetic alarms were not identified and deleted.

In test 2.3 a summary alarm was reported in KG8 in station QRS. The scenario intended this to be in the TX direction only, and to be deleted by the algorithm as a sympathetic alarm. The test results reported a fault in KG8 in the RX direction. This is because the alarm reported was a summary a alarm, a single alarm to signify an alarm in the TX direction or in the RX direction, or in both directions. The algorithm reads this as two separate alarms, one in the TX and one in the RX direction. Then the algorithm deleted the TX alarm as sympathetic and reported a communications fault in KG8 in the RX direction. Because this bidirectional alarm is treated as two separate alarms, the alarm was not totally deleted as a sympathetic alarm, even though it was reported as one.

In testing the Highly Interconnected network (described in section 5.3.2) a scenario was run in which all alarms were reported except the real fault alarm. This was intended to simulate the condition where fault information is not available from the faulted station due to degraded conditions. This is a condition that could be expected to occur, if for example the service channel should develop problems.

The algorithm does not work on a polling basis, and hence it expects all the alarm information to be correctly reported to it, as alarms occur. Because of this, when the above scenario was run, it identified the real fault as being one of the reported alarms, and did not check to see if the real fault was at a different station that had not been able to report in.

The final occurrence of incorrect results occured when running scenario A of the Backhauling network. The algorithm was unable to distinguish that a signal was being routed in and out of a terminal station on the same link. In other words, it did not recognize the backhauling connectivity, but treated the connection as a signal termination and the start of a new signal in the opposite direction. As a result the algorithm was unable to correctly delete all sympathetic alarms, although it did correctly identify the intended fault as a real fault.

6.2 Algorithm Cycle Times

Table 6-1 summarizes the average cycle time for each scenario run on each network connectivity.

Network	Scenario	Average Time
Highly Interconnected	Preliminary 1.1	131 Secs
Highly Interconnected	Preliminary 1.2	108 Secs
Highly Interconnected	Preliminary 1.3	109 Secs
Highly Interconnected	Preliminary 1.4	117 Secs
Highly Interconnected	Preliminary 1.5	111 Secs
Highly Interconnected	Preliminary 1.6	113 Secs
Highly Interconnected	Preliminary 1.7	114 Secs
Highly Interconnected	Preliminary 1.8	111 Secs
Highly Interconnected	Preliminary 1.9	114 Secs
Highly Interconnected	Preliminary 1.10	110 Secs
Highly Interconnected	A	104 Secs
Highly Interconnected	B	124 Secs
Highly Interconnected	C	116 Secs
Highly Interconnected	D	124 Secs
Highly Interconnected	E	111 Secs
Highly Interconnected	F	119 Secs
Dumbbell Structure	A	57 Secs
Dumbbell Structure	B	58 Secs
Dumbbell Structure	C	54 Secs
Dumbbell Structure	D	59 Secs
Dumbbell Structure	E	61 Secs
Loop Structure	A	19 Secs
Loop Structure	B	19 Secs
Backhauling	A	23 Secs

Table 6-2 Message Processing Times

Network	Scenario	# of MIM's	Time
Highly Interconnected	A	0	111 Secs
Highly Interconnected	A	1	123 Secs
Highly Interconnected	A	2	124 Secs
Highly Interconnected	A	4	134 Secs
Highly Interconnected	A	6	134 Secs
Highly Interconnected	B	1	123 Secs
Highly Interconnected	B	2	123 Secs
Highly Interconnected	B	4	137 Secs
Highly Interconnected	B	6	133 Secs
Highly Interconnected	C	1	122 Secs
Highly Interconnected	C	2	133 Secs
Highly Interconnected	C	4	132 Secs
Highly Interconnected	C	6	134 Secs
Highly Interconnected	D	0	111 Secs
Highly Interconnected	D	1	112 Secs
Highly Interconnected	D	2	117 Secs
Highly Interconnected	D	4	120 Secs
Highly Interconnected	D	6	125 Secs
Dumbbell Structured	A	0	52 Secs
Dumbbell Structured	A	1	62 Secs
Dumbbell Structured	A	2	57 Secs
Dumbbell Structured	A	4	60 Secs
Dumbbell Structured	A	6	61 Secs
Dumbbell Structured	B	0	56 Secs
Dumbbell Structured	B	1	60 Secs
Dumbbell Structured	B	2	62 Secs
Dumbbell Structured	B	4	62 Secs
Dumbbell Structured	B	6	61 Secs
Dumbbell Structured	C	0	59 Secs
Dumbbell Structured	C	1	55 Secs
Dumbbell Structured	C	2	60 Secs
Dumbbell Structured	C	4	65 Secs
Dumbbell Structured	C	6	62 Secs
Dumbbell Structured	D	0	50 Secs
Dumbbell Structured	D	1	60 Secs
Dumbbell Structured	D	2	65 Secs
Dumbbell Structured	D	4	61 Secs
Dumbbell Structured	D	6	58 Secs
Dumbbell Structured	E	0	55 Secs
Dumbbell Structured	E	1	57 Secs
Dumbbell Structured	E	2	57 Secs
Dumbbell Structured	E	4	61 Secs
Dumbbell Structured	E	6	60 Secs

Table 6-2 Message Processing Times (Continued)

Network	Scenario	# of MIM's	Time
Loop Structured	A	0	26 Secs
Loop Structured	A	1	22 Secs
Loop Structured	A	2	19 Secs
Loop Structured	A	4	21 Secs
Loop Structured	A	6	20 Secs
Loop Structured	B	0	20 Secs
Loop Structured	B	1	21 Secs
Loop Structured	B	2	22 Secs
Loop Structured	B	4	21 Secs
Loop Structured	B	6	20 Secs
Backhauling	A	0	17 Secs
Backhauling	A	1	16 Secs
Backhauling	A	2	16 Secs
Backhauling	A	4	21 Secs
Backhauling	A	6	20 Secs

To detect and isolate a fault in a communications network, this algorithm requires on the average two and one half cycles. This means to find a real fault in the Highly Interconnected network will require about five minutes, while in the Dumbbell structured network only about two and one half minutes is required. This shows that it is not the number of stations in a network which affects the execution time, but it is their connectivity. The fault isolation algorithm operating on the Highly Interconnected network with only five stations takes twice as long to find a fault than operating on the Dumbbell Structured network with sixteen stations.

The cycle times are very short for the Loop and Backhauling networks. These times are not useful in judging the execution time in a network since these networks are much smaller than any existing DCS networks, and were intended to be used in the accuracy testing.

The message processing tests were made to observe what effect increasing the amount of processor communications with the CPMAS-D units would have on the cycle time of the Emulator. Here each scenario was run several times, each time requesting a different number of Monitor Immediate (MIM) displays from the CPMAS-D units. As can be seen, the cycle times seemed to vary about 20% from shortest to longest execution time recorded for each scenario. As could be expected, the trend is for the execution time to increase with the increase in message processing activity.

6.3 Memory Requirements

The procedure for calculating the memory requirements of a particular network is outlined in reference (c), the Final Technical Report by GTE. The memory requirements for each network used during this testing, along with the memory requirements of the algorithm itself, are presented in table 6-3.

Table 6-3 FDIV Memory Requirements

Program or Data Name	Size (in bytes)*
FDI Algorithm	429,000
Highly Interconnected Network	84,000
Dumbbell Structured Network	85,000
Loop Network	50,000
Backhauling Network	50,000

* Always rounded up to highest 1000

6.4 Intermittent Test Results

An important predictive indicator of failures in a digital communications network is intermittent faults (Ref: FKV Pilot Digital System Evaluation, U.S. Department of Commerce, OTITS Report TM-77-238, Vol II, pp 11-12). Tests were run to determine the capability of the fault isolation algorithm to identify and display intermittent faults. The basic performance characteristics sought were; minimum time duration in which the equipment must be alarmed in order to be identified, and what is the maximum rate of occurrence which can be counted?

Since the objective of these tests is concerned with timing, they were performed on a single network, rather than all four different networks. We do not expect a great deal of difference in performance as the network topology is varied. Since the highly interconnected network represents the expected "worst case" in terms of algorithm execution time, this network was used for the testing of intermittents.

For each class of fault, (binary, analog, and pulse) the rate of occurrence and duration of fault was varied over a wide range and system response was observed. Faults were inserted by manually switching alarms on and off the monitor point simulators.

The intermittent fault data is presented in figures 6.7-1 through 6.7-3. Figures 6.7-1 and 6.7-2 deal with the minimum time duration for which the equipment must be alarmed in order to be identified. Figure 6.7-1 graphically shows single faults that had not previously been recognized, and includes the minimum time durations used, and whether or not the algorithm recognized the fault. Each fault duration time was repeated 10 times. There is no distinct "minimum fault duration time," where lesser fault durations are ignored and greater fault durations are recognized. But there is a general trend and it can be said that over 50% of faults of at least 45 seconds in duration will be recognized, and over 95% of faults of at least 45 seconds in duration will be recognized, and over 95% of faults of 60 seconds or more duration will be recognized.

Figure 6.7-2 deals with single faults that have not been cleared from the CPMAS display. On the display, active faults are tagged as "in", non-active faults that have not been cleared are tagged as "out". Here the minimum time a fault has to be active is essentially in the 10 to 20 second range, and again there is no distinct "minimum faut durational time."

Figure 6.7-3 deals with the maximum rate of occurrence which can be counted. Faults were switched on and off, 50% of the time on, and 50% of the time off, and this sequence repeated 10 times. From the CPMAS display column labeled "cnt", we then ascertain the number of fault-on times the algorithm recognized. From the figure we see that as the time durations decrease below 4 seconds, the percent of error increases to over 90%. The algorithm recognizes only 1 or 2 fault-on times, and not the 10 that were inserted.

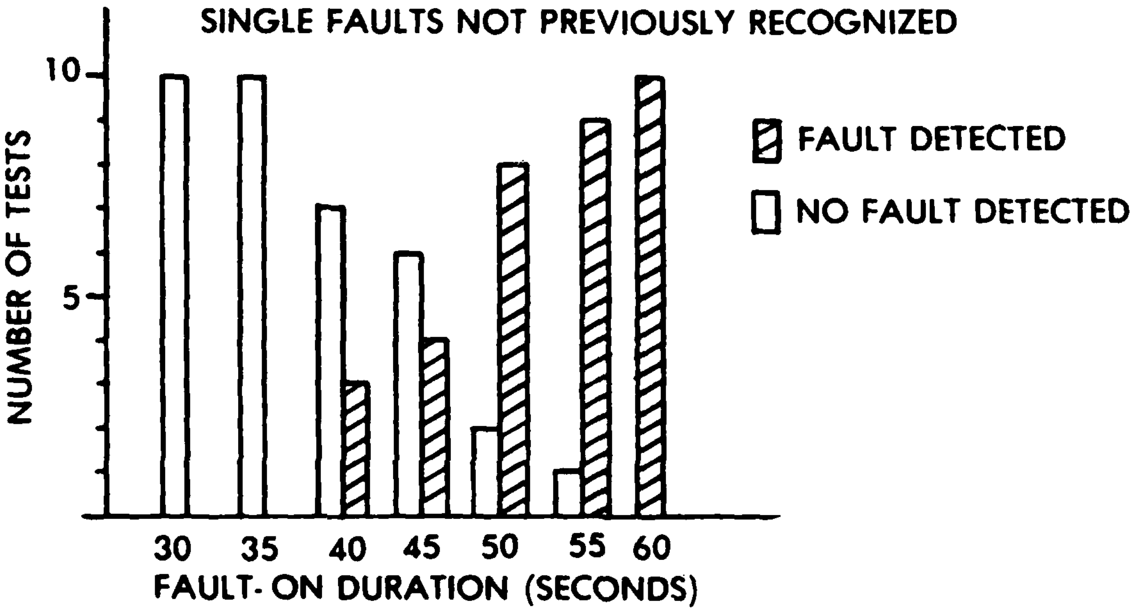


FIGURE 6.7-1

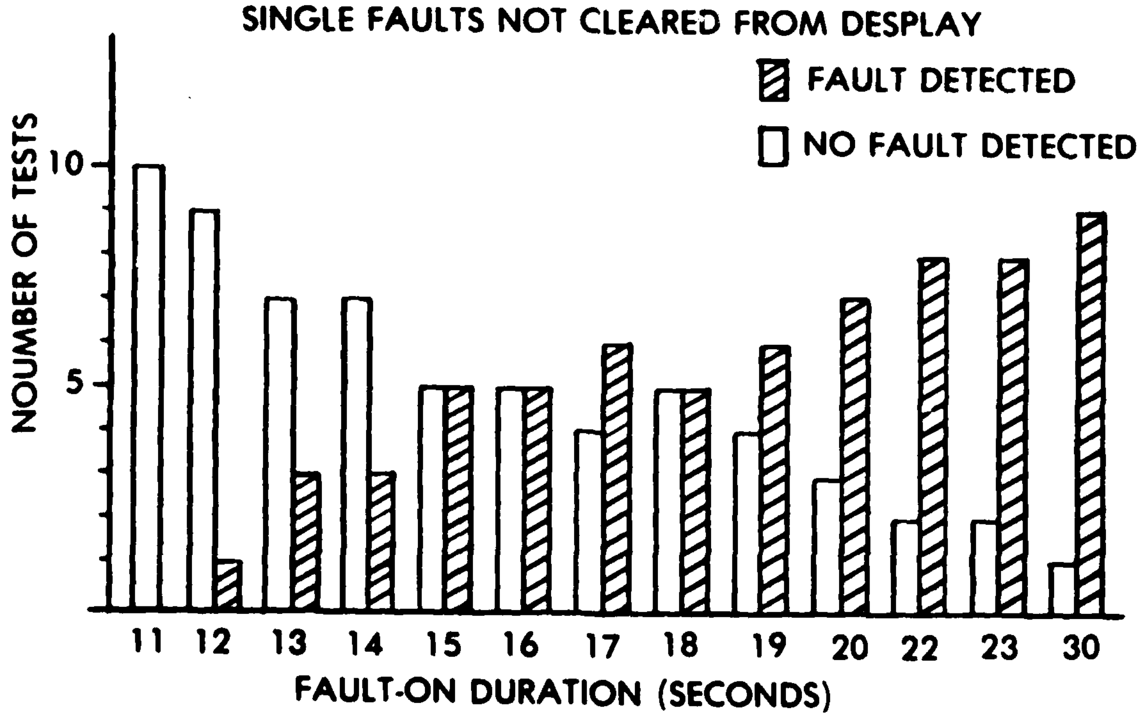


FIGURE 6.7-2

RATE OF FAULT OCCURENCE

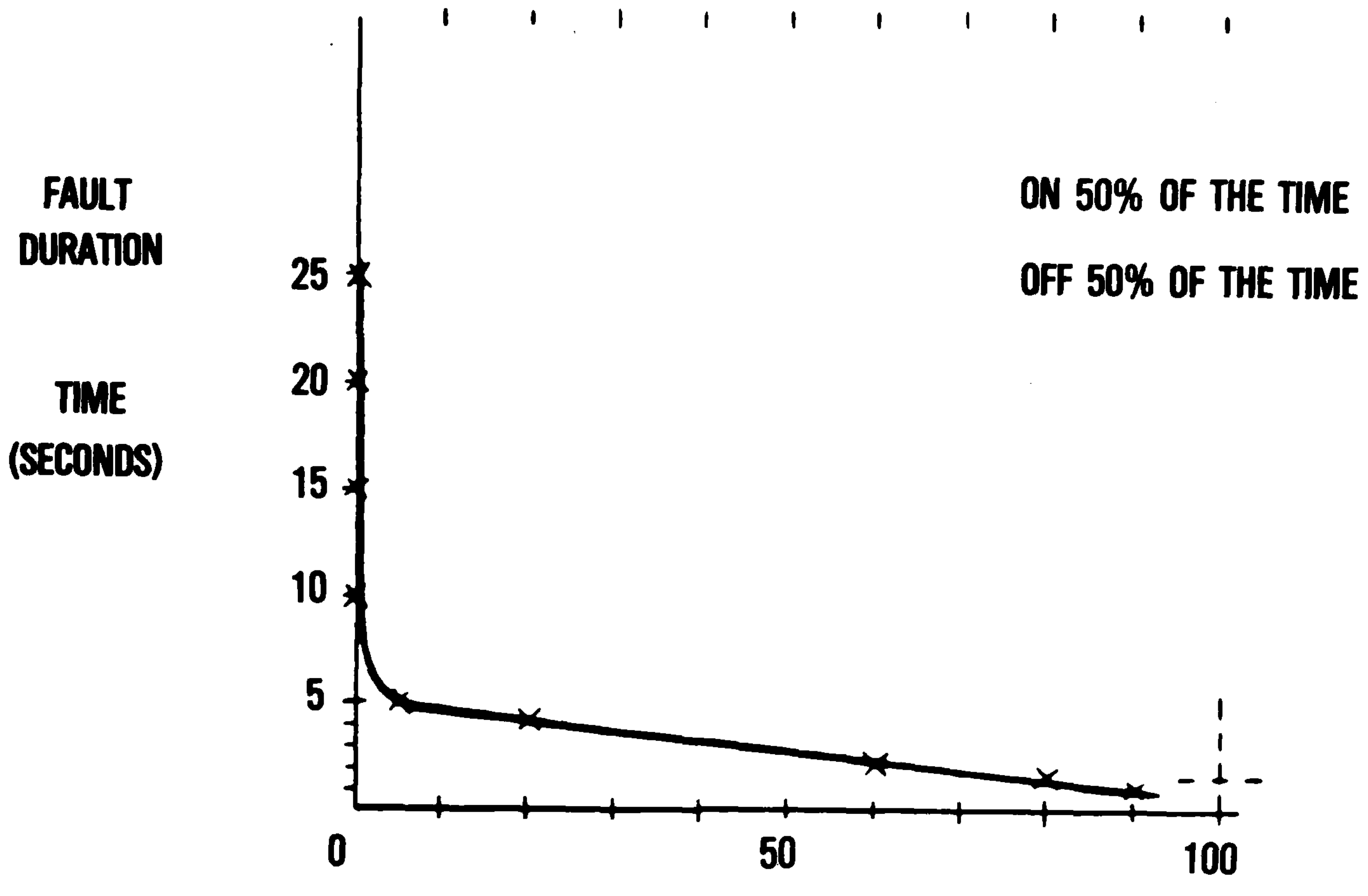


FIGURE 6.7-3

7.0 Recommendations

Based upon the results of testing, the fault isolation algorithm works very well, however there are several minor problems which need to be corrected in the next version of this algorithm. The recommended corrections to the algorithm are presented in this section:

a) As was demonstrated in the testing, the algorithm cycle time is highly dependent on the connectivity of the test network. In their final report, GTE has done an extensive analysis measuring this dependence. Basically they say that the time increases in proportion to the number of subnode center stations squared. Subnode center stations are specially defined stations by the user when he creates the network connectivity files. The CPMAS emulator requires that any station with 3 or more links connected to it be defined as a subnode center.

This time dependence is a definite problem in planned application of the fault detection/isolation algorithm. It is planned that the algorithm will receive alarm information from several stations over a wide geographical area, since it will most likely reside at a high level in the DCS hierarchy. This means that the network connectivity will have several subnode center stations in it, and as a result the fault isolation cycle time could become quite large. It will certainly be more than 45 secs, which is the time required to meet the initial design specification of a 2 minute fault isolation time.

In order to meet the design requirement of a maximum fault isolation time of 2 minutes, it will be necessary to eliminate the n squared time dependence which the algorithm currently has. This will require using a different or modified search technique than the algorithm currently uses to map fault connectivity in the network.

b) Since this system was a prototype, not much effort was placed in the man to machine interface, other than to make it functional. Since the system will be used by DCS tech controllers who by necessity receive minimal on-the-job training, every effort must be made to make the system easy to use and to provide as much useful information in each display as possible without confusing the operator. Current work in man/machine interface should be reviewed, and where applicable, used.

c) The weakest part of the CPMAS emulator is the data base. This consists of the data base generator, and the actual data structure itself. Currently, the user must input network and station connectivity data into the data base manually. However, this information is available through other data bases, on other computer systems. The data base generator should be able to read these connectivity data bases, and then either use these files directly, or translate them into a format which the algorithm can use. The current system is a very primitive data base generator with minimum error diagnostis. Future data base generation should provide explicit error messages to the user to alert him to problems. A means for allowing the user to correct erroneous connectivity data manually should be provided. Since this information is vital to the performance of the algorithm, means of protection should be incorporated into the system. This could mean protected access to certain types of files, and password access to the system.

In addition to connectivity information, equipment and monitor point information is also kept in the data base. Currently the data base contains rigid equipment types which cannot be changed except by physically rewriting portions of the software. This does not lend itself to easy modification if a new piece of equipment is added to the station to replace an older type, or is a new type of equipment. There should be a way for each station to define the charac⁴ ⁊ristics of each piece of equipment it has to the algorithm, and should this equipment change, then all that is necessary is to input the new equipments characteristics over tty console to the emulator.

The monitor points used in this test program are not representative of the monitor points actually used. In fact, the monitor points available to the fault isolation algorithm when it will be ready to be tested in the field may not be the same ones used now. Some method should be used to allow the monitor point information to be redefined to the algorithm each time it is changed. This will present a problem, since the less monitor points available to the algorithm, the harder it is to correctly isolate the fault. This may require redefining the assumptions the algorithm makes according to the alarm information it receives as to where the real fault(s) are. Although there may be no easy solution to this problem, it should certainly be looked into and some provisions should be made to handle future monitor point changes.

d) The way the algorithm handles intermittent faults could be improved. Right now, there is a window of uncertainty in which if an intermittent fault becomes active, and then inactive, it may not be detected. This is because the algorithm only keeps track of the latest status of each fault reported. The portion of the cycle which the algorithm is in determines how long it will be before the algorithm checks to see the current status of each alarm. It was demonstrated that an alarm may be active for a few seconds, and then go inactive again and if the timing is correct, the algorithm will have never detected the alarm. It was also demonstrated that if the alarm is active for at least one cycle time (maximum time necessary) then it will always be detected by the algorithm. For a 2 minute fault isolation time, this puts a requirement that for an intermittent alarm to be detected it must have a 45 second transition time (from fault to no-fault or from no-fault to fault).

Although this may be acceptable for most intermittent faults, the time necessary in which an intermittent fault must be active to be detected can be reduced by keeping track of each reported occurrence of a particular fault independent of the fault isolation cycle. Then when the algorithm reaches the point in the cycle where it looks for reported alarms, it will receive a complete history for an intermittent alarm.

e) It may be necessary to have some kind of intelligent polling scheme added to the algorithm to overcome the weakness of assuming that no received fault information from a station indicates that the station is working correctly. As was mentioned before, if the service channel is inoperative, then this assumption is invalid. A solution to this would be a scheme where the algorithm would check with stations it did not receive alarm information from. If this normally requires a lot of time polling stations that just did not have fault information to report, this time could be reduced by polling only those stations suspect, such as stations along a current propagation path.

f) Finally, there is clearly a problem with the algorithm in trying to correctly delete all sympathetic alarms in the case of a backhauling connectivity. At this point we are not sure if this is a fault of the algorithm, or a fault of the data base generator. However, this problem should be looked into, and the solution should be incorporated into the next version of this algorithm.

ATE

LME

8